

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

SSSSSSSS	YY	YY	MM	MM	BBBBBBBB	000000	LL	
SSSSSSSS	YY	YY	MM	MM	BBBBBBBB	000000	LL	
SS	YY	YY	MMM	MMM	BB	00	LL	
SS	YY	YY	MMM	MMM	BB	00	LL	
SS	YY	YY	MM	MM	BB	00	LL	
SS	YY	YY	MM	MM	BB	00	LL	
SSSSSS	YY		MM	MM	BBBBBBBB	00	LL	
SSSSSS	YY		MM	MM	BBBBBBBB	00	LL	
SS	YY		MM	MM	BB	00	LL	
SS	YY		MM	MM	BB	00	LL	
SS	YY		MM	MM	BB	00	LL	
SSSSSSSS	YY		MM	MM	BB	00	LL	
SSSSSSSS	YY		MM	MM	BBBBBBBB	000000	LLLLLLLLLL
					BBBBBBBB	000000	LLLLLLLLLL

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS

SYMBOL
Table of contents

- SYMBOL TABLE MANIPULATION ROUTINES^{N 4}

16-SEP-1984 00:22:24 VAX/VMS Macro V04-00

Page 0

(3)	73	ALLOCATE AND INSERT ENTRY IN SYMBOL TABLE
(5)	219	DEALLOCATE SYMBOL TABLE ENTRY
(6)	254	CONVERT SYMBOL VALUE TO STRING
(7)	291	CONVERT EXPRESSION RESULT TO STRING
(8)	328	SEARCH FOR SYMBOL ENTRY
(9)	366	SEARCH FOR SYMBOL ENTRY IN A LOCAL SYMBOL TABLE
(10)	427	SEARCH FOR SYMBOL ENTRY IN GLOBAL SYMBOL TABLE
(11)	458	SEARCH SPECIFIC SYMBOL TABLE FOR ENTRY
(12)	510	RESTORE SYMBOL DEFINITION AFTER A SPAWN
(13)	582	DELETE SYMBOL FROM SYMBOL TABLE

```
0000 1 .TITLE SYMBOL - SYMBOL TABLE MANIPULATION ROUTINES
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28 SYMBOL TABLE MANIPULATION ROUTINES
0000 29
0000 30 D. N. CUTLER 29-APR-77
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-006 HWS0031 Harold Schultz 14-Mar-1984
0000 35 Add DELETE/SYMBOL/LOG
0000 36
0000 37 V03-005 PCG0009 Peter George 16-Aug-1983
0000 38 Fix bug in binary symbol restoration logic.
0000 39
0000 40 V03-004 PCG0008 Peter George 27-May-1983
0000 41 Add PTRDEF reference.
0000 42
0000 43 V03-003 PCG0007 Peter George 27-May-1983
0000 44 Add DCL$DELSYM.
0000 45
0000 46 V03-002 PCG0006 Peter George 09-Mar-1983
0000 47 Call DCL$FIND KEYPAD.
0000 48 Add DCL$RESTORE_SYM.
0000 49
0000 50 V03-001 PCG0005 Peter George 15-Nov-1982
0000 51 Do roundup in DEADYNMEM. Signal SPR error.
0000 52 :---
```



```
0000 54 :  
0000 55 : MACRO LIBRARY CALLS  
0000 56 :  
0000 57 : PRCDEF ;DEFINE PROCESS WORK AREA  
0000 58 : WRKDEF ;DEFINE COMMAND WORK AREA  
0000 59 : PTRDEF ;DEFINE TOKEN DESCRIPTORS  
0000 60 : SYMDEF ;DEFINE SYMBOL ENTRY OFFSETS  
0000 61 : IDFDEF ;DEFINE INDIRECT STACK OFFSETS  
0000 62 : CTXDEF ;DEFINE SPAWN CTX SYMBOLS  
0000 63 : $CLIMSGDEF ;DEFINE ERROR/STATUS VALUES  
0000 64 :  
00000000 65 : .PSECT DCL$ZCODE,BYTE,RD,NOWRT  
0000 66 :  
0000 67 : SYMBOL TYPE DESCRIPTION STRINGS FOR DEL/SYM/LOG  
0000 68 :  
6C 61 63 6F 4C 00' 0000 69 LOCTAB: .ASCIC /Local/  
05 0000  
0006  
6C 61 62 6F 6C 47 00' 0006 70  
06 0006 71 GBLTAB: .ASCIC /Global/
```

```
000D 73 .SBTTL ALLOCATE AND INSERT ENTRY IN SYMBOL TABLE
000D 74 :+
000D 75 : DCL$ALLOCSYMABR - ALLOCATE AND INSERT ENTRY IN SYMBOL TABLE W/ ABBREVIATION
000D 76 : DCL$ALLOCSYM - ALLOCATE AND INSERT ENTRY IN SYMBOL TABLE
000D 77 :
000D 78 : THIS ROUTINE IS CALLED TO ALLOCATE AND INSERT A SYMBOL ENTRY IN EITHER THE
000D 79 : LOCAL OR GLOBAL SYMBOL TABLE.
000D 80 :
000D 81 : INPUTS:
000D 82 :
000D 83 : R11 = ADDRESS OF PROCESS WORK AREA
000D 84 :
000D 85 : R0 = TYPE OF SYMBOL VALUE (SYM_K_STRING OR SYM_K_BINARY)
000D 86 : R1/R2 = DESCRIPTOR OF SYMBOL VALUE
000D 87 : R3/R4 = DESCRIPTOR OF SYMBOL NAME.
000D 88 : R5 = ADDRESS OF SYMBOL TABLE LISTHEAD.
000D 89 :
000D 90 : IT IS ASSUMED THAT CONTROL Y/C AST'S ARE DISABLED.
000D 91 :
000D 92 : OUTPUTS:
000D 93 :
000D 94 : THE SPECIFIED SYMBOL TABLE IS SEARCHED FOR THE SPECIFIED ENTRY, AND
000D 95 : IF FOUND, THE OLD ENTRY IS DEALLOCATED. A SYMBOL TABLE ENTRY IS THEN
000D 96 : ALLOCATED, FILLED WITH THE SYMBOL AND VALUE INFORMATION, AND THEN
000D 97 : INSERTED IN THE SPECIFIED SYMBOL TABLE.
000D 98 :
000D 99 : R0 LOW BIT CLEAR INDICATES ALLOCATION FAILURE WITH:
000D 100 :
000D 101 : R0 = DCL$_SYMOVF - NO ROOM FOR SYMBOL DEFINITIONS.
000D 102 : R0 = DCL$_SYMDEL - ABBREVIATED SYMBOL NOT ALLOWED.
000D 103 : R0 = DCL$_ABSYMD - AMBIGUOUS SYMBOL DEFINITION.
000D 104 :
000D 105 : R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
000D 106 :
000D 107 : R1 = ADDRESS OF ALLOCATED SYMBOL ENTRY.
000D 108 : R2,R3,R4,R5 ARE DESTROYED.
000D 109 : -
000D 110 :
000D 111 DCL$GT_SYMABR:: : ABBREVIATED SYMBOL COUNTED STRING
000D 112 : .ASCIC '*'
000D 113 :
000D 114 DCL$ALLOCSYMABR:: : ALLOCATE AND INSERT SYMBOL IN TABLE
000D 115 : #M<R0,R1,R2,R3,R4,R5> : SAVE SYMBOL ENTRY PARAMETERS
000D 116 : #A'*,R3,(R4) : FIND THE ABBREVIATION FLAG
000D 117 : R0 : SAVE NUMBER OF EXTRA CHARACTERS
000D 118 : ALLOCSYM : BRANCH IF NOT ABBREVIATED
000D 119 : (SP) : REMOVE THE ABBR CHAR FROM EXTRA COUNT
000D 120 : 16(SP) : REMOVE THE ABBR CHAR FROM NAME COUNT
000D 121 : (SP),1(R1),(R1) : SLIDE REMAINING CHARACTERS OVER IT
000D 122 : BRB ALLOCSYM
000D 123 :
000D 124 DCL$ALLOCSYM:: : ALLOCATE AND INSERT SYMBOL IN TABLE
000D 125 : #M<R0,R1,R2,R3,R4,R5> : SAVE SYMBOL ENTRY PARAMETERS
000D 126 : -(SP) : SET NO ABBREVIATION
000D 127 : #A'*,R3,(R4) : FIND THE ABBREVIATION FLAG
000D 128 : BEQL ALLOCSYM : OK IF NONE FOUND
```

64 53 3F BB 000F 115
2A 3A 0011 116
50 DD 0015 117
22 13 0017 118
6E D7 0019 119
AE D7 001B 120
61 01 A1 10 6E 28 001E 121
16 11 0023 122
0025 123
0025 124
3F BB 0025 125
7E D4 0027 126
64 53 2A 3A 0029 127
0C 13 002D 128


```
50 5E 1C AE 9E 002F 129 MOVAB 7*4(SP),SP ;CLEAN STACK
    00038278 8F D0 0033 130 MOVL #CLIS_SYMABR,R0 ;SET NO SYMBOL ABBREVIATIONS STATUS
    05 003A 131 RSB
    003B 132
    003B 133
    003B 134
    003B 135
    003B 136
    003B 137
    003B 138
    003B 139
    003B 140
    003B 141
    51 55 D4 003B 141 CLRL R5 ;START WITH FULL SYMBOL NAME
    10 AE 7D 003D 142 10$: MOVQ 16(SP),R1 ;RESET SYMBOL PARAMETERS
    51 55 C2 0041 143 SUBL R5,R1 ;FIND SIZE TO SEARCH FOR THIS TIME
    50 18 AE D0 0044 144 MOVL 24(SP),R0 ;SET ADDRESS OF SYMBOL TABLE LISTHEAD
    013E 30 0048 145 BSBW DCL$SEARCHT ;SEARCH FOR SYMBOL
    11 50 E9 004B 146 BLBC R0,30$ ;IF LBC SEARCH FAILURE
    01 0A A3 91 004E 147 CMPB SYM_B_TYPE(R3),#SYM_K_PERM ;PERMANENT SYMBOL?
    74 13 0052 148 BEQL SYMOVF ;IF YES - REFUSE TO ALLOCATE
    0054 149 ASSUME SYM_B_NONUNIQUE EQ SYM_T_SYMBOL-1
    51 74 51 83 0054 150 SUBB3 R1,-(R4),R1 ;FIND SYMBOL'S NEW
    51 97 0058 151 DECB R1 ;UNIQUENESS POINT
    63 1E 005A 152 BGEQU ABSYMD ;IF GEQ, AMBIGUOUS SYMBOL
    0086 30 005C 153 BSBW DCL$DEALLOCOSYM ;DEALLOCATE SYMBOL ENTRY
    55 6E F3 005F 154 30$: AOBLEQ (SP),R5,10$ ;LOOP IF MORE SYMBOLS TO CHECK
    0063 155
    51 10 AE 08 AE C1 0063 156 ADDL3 8(SP),16(SP),R1 ;CALCULATE LENGTH OF SYMBOL STRINGS
    02 04 AE D1 0069 157 CMPL 4(SP),#SYM_K_BINARY ;BINARY VALUE?
    05 12 006D 158 BNEQ 40$ ;BRANCH IF NOT
    51 10 AE 04 C1 006F 159 ADDL3 #4,16(SP),R1 ;SET LENGTH OF SYMBOL PLUS LONGWORD
    51 0F C0 0074 160 40$: ADDL #SYM_T_SYMBOL+3,R1 ;ADD IN FIXED OVERHEAD AND ROUND
    4C 50 E9 0077 161 BSBW DCL$ALCDYNMEM ;ALLOCATE DYNAMIC MEMORY
    08 A2 51 D0 007C 162 BLBC R0,SYMOVF ;IF LBC ALLOCATION FAILURE
    08 A2 8E F6 0080 163 MOVL R1,SYM_W_SIZE(R2) ;SET SIZE OF ALLOCATED BLOCK, ETC.
    0A A2 8E F6 0084 164 CVTLB (SP)+,SYM_B_NONUNIQUE(R2) ;SET UNIQUENESS POINT
    10 BE 62 0E 0088 165 CVTLB (SP)+,SYM_B_TYPE(R2) ;SET SYMBOL VALUE TYPE
    10 AE 52 D0 008C 166 INSQUE SYM_L_FL(R2),@16(SP) ;INSERT ENTRY IN SYMBOL TABLE
    53 08 AE 7D 0090 167 MOVL R2,T6(SP) ;SAVE ADDRESS OF NEW ENTRY
    0C A2 53 90 0094 168 MOVQ 8(SP),R3 ;GET SYMBOL NAME
    0D A2 64 53 28 0098 169 MOVQ R3,SYM_T_SYMBOL(R2) ;INSERT LENGTH OF SYMBOL
    51 6E 7D 009D 170 MOVC R3,(R4),SYM_T_SYMBOL+1(R2) ;INSERT SYMBOL NAME
    50 10 AE D0 00A0 171 MOVQ (SP),R1 ;GET SYMBOL VALUE
    02 0A A0 91 00A4 172 MOVL 16(SP),R0 ;RETRIEVE ADDRESS OF ENTRY
    05 12 00A8 173 CMPB SYM_B_TYPE(R0),#SYM_K_BINARY ;BINARY VALUE?
    83 51 D0 00AA 174 BNEQ 50$ ;BRANCH IF STRING VALUE
    07 11 00AD 175 MOVL R1,(R3)+ ;STORE LONGWORD BINARY VALUE
    83 51 B0 00AF 176 BRB 60$
    63 62 51 28 00B2 177 50$: MOVW R1,(R3)+ ;INSERT LENGTH OF STRING VALUE
    3E BA 00B6 178 MOVC R1,(R2),(R3) ;INSERT STRING VALUE
    51 55 D0 00B8 179 60$: POPR #^M<R1,R2,R3,R4,R5> ;RESTORE REGISTERS
    50 01 D0 00BB 180 MOVL R5,R1 ;RETURN ADDRESS OF SYMBOL ENTRY
    05 00BE 181 MOVL #1,R0 ;SET SUCCESS INDICATOR
    00BF 182
    00BF 183
    00BF 184
    00BF 185
    .ENABL LSB
```

SYMBOL
V04-000

F 5
- SYMBOL TABLE MANIPULATION ROUTINES
ALLOCATE AND INSERT ENTRY IN SYMBOL TABL

16-SEP-1984 00:22:24 VAX/VMS Macro V04-00
4-SEP-1984 23:43:47 [DCL.SRC]SYMBOL.MAR;1

Page 5
(3)

07	11	00BF	186	ABSYMD: STATUS	ABSYMD		
		00C3	187	BRB	90\$;SET AMBIGUOUS SYMBOL STATUS
		00C8	188				
5E	08	00C8	189	SYMOVF: STATUS	SYMOVF		;SET SYMBOL TABLE OVERFLOW STATUS
	3E	00CF	190	90\$: ADDL	#8,SP		;POP OFF TOP 2 LONGWORDS
		00D2	191	POPR	#^M<R1,R2,R3,R4,R5>		;RESTORE REGISTERS
		00D4	192	RSB			
		00D5	193				
		00D5	194	.DSABL	LSB		

- SYMBOL TABLE MANIPULATION ROUTINES 16-SEP-1984 00:22:24 VAX/VMS Macro V04-00
 ALLOCATE AND INSERT ENTRY IN SYMBOL TABL 4-SEP-1984 23:43:47 [DCL.SRC]SYMBOL.MAR;1

Page 6
(4)

[illegible]

```
00E5 219 .SBTTL DEALLOCATE SYMBOL TABLE ENTRY
00E5 220 :+
00E5 221 : DCL$DEALLOCSYM - DEALLOCATE SYMBOL TABLE ENTRY
00E5 222 : DCL$DEADYNMEM - DEALLOCATE DYNAMIC MEMORY
00E5 223 :
00E5 224 : THIS ROUTINE IS CALLED TO REMOVE A SYMBOL FROM ITS TABLE AND/OR
00E5 225 : DEALLOCATE DYNAMIC MEMORY USED.
00E5 226 :
00E5 227 : INPUTS TO DEALLOCSYM:
00E5 228 :
00E5 229 : R11 = ADDRESS OF PROCESS WORK AREA
00E5 230 : R3 = ADDRESS OF SYMBOL ENTRY.
00E5 231 :
00E5 232 : INPUTS TO DEALDYNMEM:
00E5 233 :
00E5 234 : R11 = ADDRESS OF PROCESS WORK AREA
00E5 235 : R0 = ADDRESS OF BLOCK TO DEALLOCATE
00E5 236 : R1 = SIZE OF BLOCK
00E5 237 :
00E5 238 : IT IS ASSUMED THAT CONTROL Y/C AST'S ARE DISABLED.
00E5 239 :
00E5 240 : OUTPUTS:
00E5 241 :
00E5 242 : THE SYMBOL IS REMOVED FROM ITS TABLE AND/OR ITS STORAGE IS DEALLOCATED.
00E5 243 :-
00E5 244 :
00E5 245 DCL$DEALLOCSYM::
51 50 63 0F 00E5 246 REMQUE SYM_L_FL(R3),R0 ;DEALLOCATE SYMBOL TABLE ENTRY
51 08 A0 3C 00E8 247 MOVZWL SYM_W_SIZE(R0),R1 ;REMOVE SYMBOL ENTRY FROM ITS TABLE
00EC 248 DCL$DEADYNMEM:: ;GET SIZE OF BLOCK TO DEALLOCATE
00EC 249 ADDL #7,R1 ;DEALLOCATE DYNAMIC MEMORY
51 07 07 C0 00EC 250 BICL #7,R1 ;ROUND UP TO QUADWORD BOUNDARY
51 07 07 CA 00EF 251 MOVAB PRG_Q_ALLOCREG(R11),R3 ;TRUNCATE TO QUADWORD MULTIPLE
53 20 AB 9E 00F2 251 ;GET ADDRESS OF ALLOCATION LISTHEAD
00000000'9F 17 00F6 252 JMP @#EXE$DEALLOCATE ;DEALLOCATE SYMBOL ENTRY STORAGE
```



```
00FC 254 .SBTTL CONVERT SYMBOL VALUE TO STRING
00FC 255 :+
00FC 256 : DCL$SYM_STRING - GET SYMBOL VALUE AND CONVERT TO A STRING
00FC 257 :
00FC 258 : THIS ROUTINE IS CALLED TO SEARCH ALL LOCAL AND GLOBAL SYMBOL TABLES FOR
00FC 259 : A SPECIFIED SYMBOL, AND TO RETURN THE STRING FORM OF THE SYMBOL VALUE.
00FC 260 : THAT IS, IF THE SYMBOL VALUE IS NUMERIC, IT IS CONVERTED TO ASCII DECIMAL
00FC 261 : BEFORE BEING RETURNED.
00FC 262 :
00FC 263 : INPUTS:
00FC 264 :
00FC 265 : R11 = ADDRESS OF PROCESS WORK AREA
00FC 266 :
00FC 267 : R1 = LENGTH OF SYMBOL.
00FC 268 : R2 = ADDRESS OF SYMBOL.
00FC 269 :
00FC 270 : OUTPUTS:
00FC 271 :
00FC 272 : R0 = STATUS
00FC 273 : R1 = LENGTH OF VALUE STRING
00FC 274 : R2 = ADDRESS OF VALUE STRING
00FC 275 : R3 = DESTROYED
00FC 276 : R4 = TABLE FLAG
00FC 277 : 1 ==> FOUND IN LOCAL SYMBOL TABLE
00FC 278 : 2 ==> FOUND IN GLOBAL SYMBOL TABLE
00FC 279 :
00FC 280 : THE STRING MAY ACTUALLY RESIDE IN EITHER THE SYMBOL TABLE OR
00FC 281 : IN THE UNUSED PORTION OF THE EXPANSION BUFFER. IT IS THE CALLER'S
00FC 282 : RESPONSIBILITY TO COPY THE STRING BEFORE USING THE EXPANSION BUFFER.
00FC 283 :-
00FC 284 :
00FC 285 DCL$SYM_STRING::
03 19 10 00FC 286 BSBB DCL$SEARCH ; SEARCH ALL SYMBOL TABLES
50 E8 00FE 287 BLBS R0,DCL$CVT_STRING ; BRANCH IF NOT FOUND
51 D4 0101 288 CLRL R1 ; RETURN NULL STRING ON ERROR
05 0103 289 RSB
```

```
0104 291 .SBTTL CONVERT EXPRESSION RESULT TO STRING
0104 292 :+
0104 293 : DCL$CVT_STRING - CONVERT EXPRESSION RESULT TO A STRING
0104 294 :
0104 295 : IF THE SYMBOL VALUE IS NUMERIC, IT IS CONVERTED TO ASCII DECIMAL.
0104 296 :
0104 297 : INPUTS:
0104 298 :
0104 299 : R11 = ADDRESS OF PROCESS WORK AREA
0104 300 :
0104 301 : R1/R2 = QUADWORD DESCRIBING VALUE:
0104 302 : IF R2 NONZERO, QUADWORD IS A STRING DESCRIPTOR
0104 303 : IF R2 ZERO, R1 IS A BINARY LONGWORD VALUE
0104 304 :
0104 305 : OUTPUTS:
0104 306 :
0104 307 : R0 = STATUS
0104 308 : R1 = LENGTH OF VALUE STRING
0104 309 : R2 = ADDRESS OF VALUE STRING
0104 310 : R3 = DESTROYED
0104 311 : R4 = PRESERVED
0104 312 :
0104 313 : THE STRING MAY ACTUALLY RESIDE IN EITHER THE SYMBOL TABLE OR
0104 314 : IN THE UNUSED PORTION OF THE EXPANSION BUFFER. IT IS THE CALLER'S
0104 315 : RESPONSIBILITY TO COPY THE STRING BEFORE USING THE EXPANSION BUFFER.
0104 316 :-
0104 317
0104 318 DCL$CVT_STRING::
0104 319 TSTL R2 ; NUMERIC VALUE?
0104 320 BNEQ 90$ ; BRANCH IF STRING
0104 321 MOVL R1,R0 ; GET BINARY VALUE
0104 322 PUSHL R4 ; SAVE R4 (JUST IN CASE)
0104 323 BSBW DCL$CBTA DEC ; CONVERT TO ASCII IN EXPANSION BUFFER
0104 324 MOVL (SP)+, R4 ; RESTORE SAVED R4
0104 325 90$: MOVL #1,R0 ; SET SUCCESS
0104 326 RSB
```

50 52 D5 0104 319
0B 12 0106 320
51 D0 0108 321
54 DD 010B 322
FEF0' 30 010D 323
54 8E D0 0110 324
50 01 D0 0113 325
05 0116 326


```
0117 328 .SBTTL SEARCH FOR SYMBOL ENTRY
0117 329 :+
0117 330 : DCL$SEARCH - SEARCH FOR SYMBOL ENTRY
0117 331 :
0117 332 : THIS ROUTINE IS CALLED TO SEARCH ALL LOCAL AND GLOBAL SYMBOL TABLES FOR
0117 333 : AN ENTRY. THE LOCAL SYMBOL TABLES FOR EACH OF THE INDIRECT LEVELS ARE
0117 334 : FIRST SEARCHED. IF NOTHING FOUND, THE GLOBAL SYMBOL TABLE IS SEARCHED.
0117 335 :
0117 336 : INPUTS:
0117 337 :
0117 338 : R11 = ADDRESS OF PROCESS WORK AREA
0117 339 :
0117 340 : R1 = LENGTH OF SYMBOL.
0117 341 : R2 = ADDRESS OF SYMBOL.
0117 342 :
0117 343 : OUTPUTS:
0117 344 :
0117 345 : R0 = STATUS
0117 346 : R1/R2 = QUADWORD DESCRIBING SYMBOL VALUE:
0117 347 : IF R2 NONZERO, QUADWORD IS A STRING DESCRIPTOR
0117 348 : IF R2 ZERO, R1 IS A BINARY LONGWORD VALUE
0117 349 :
0117 350 : R3 = ADDRESS OF SYMBOL ENTRY.
0117 351 : R4 = TABLE FLAG
0117 352 : 1 ==> FOUND IN LOCAL SYMBOL TABLE
0117 353 : 2 ==> FOUND IN GLOBAL SYMBOL TABLE
0117 354 : -
0117 355 :
0117 356 DCL$SEARCH::
0117 357 BSBW DCL$SEARCH_LOCAL :SEARCH FOR SYMBOL ENTRY
54 01 D0 011A 358 MOVL #1,R4 :SEARCH LOCAL SYMBOL TABLES
0B 50 E8 011D 359 BLBS R0,10$ :INDICATE LOCAL SYMBOL TABLE MATCH FOUND
005A 30 0120 360 BSBW DCL$SEARCH_GLOBAL :IF LBS MATCH FOUND
54 02 D0 0123 361 MOVL #2,R4 :SEARCH GLOBAL SYMBOL TABLE
02 50 E8 0126 362 BLBS R0,10$ :INDICATE GLOBAL SYMBOL TABLE MATCH FOUND
51 D4 0129 363 CLRL R1 :IF LBS MATCH FOUND
05 012B 364 10$: RSB :RETURN NULL STRING IF NO MATCH
:
```

```
012C 366 .SBTTL SEARCH FOR SYMBOL ENTRY IN A LOCAL SYMBOL TABLE
012C 367 :+
012C 368 :DCL$SEARCH_LOCAL - SEARCH FOR SYMBOL ENTRY IN A LOCAL SYMBOL TABLE
012C 369 :
012C 370 :THIS ROUTINE IS CALLED TO SEARCH ALL LOCAL SYMBOL TABLES FOR AN ENTRY.
012C 371 :THE LOCAL SYMBOL TABLES FOR EACH OF THE INDIRECT LEVELS ARE SEARCHED.
012C 372 :
012C 373 :INPUTS:
012C 374 :
012C 375 :R11 = ADDRESS OF PROCESS WORK AREA
012C 376 :
012C 377 :R1 = LENGTH OF SYMBOL.
012C 378 :R2 = ADDRESS OF SYMBOL.
012C 379 :
012C 380 :OUTPUTS:
012C 381 :
012C 382 :R0 = STATUS
012C 383 :R1/R2 = QUADWORD DESCRIBING SYMBOL VALUE:
012C 384 :IF R2 NONZERO, QUADWORD IS A STRING DESCRIPTOR
012C 385 :IF R2 ZERO, R1 IS A BINARY LONGWORD VALUE
012C 386 :
012C 387 :R3 = ADDRESS OF SYMBOL ENTRY.
012C 388 :-
012C 389 :
012C 390 DCL$SEARCH_LOCAL::
012C 391 :DISABLE
012C 392 :
0132 393 :SEARCH FOR SYMBOL ENTRY IN LOCAL TABLE
0132 394 :DISABLE CTRL/Y'S TO GUARANTEE INTEGRITY
0132 395 :OF PRC Q LOCAL(R11)
0132 396 :SAVE CURRENT LOCAL SYMBOL TABLE LISTHEAD
0132 397 :SAVE ADDRESS OF INDIRECT STACK POINTER
0132 398 :SET ADDRESS OF LOCAL SYMBOL TABLE LISTHEAD
0132 399 :SEARCH LOCAL SYMBOL TABLE FOR ENTRY
0132 400 :IF LBS MATCH FOUND
0132 401 :RETRIEVE ADDRESS OF INDIRECT FRAME
0132 402 :CALCULATE ADDRESS OF NEXT FRAME
0132 403 :REPLACE LISTHEAD
0132 404 :IF SET, SEARCH NEXT TABLE
0132 405 :NO MATCH FOUND
0132 406 :
0132 407 :GET THE SYMBOL VALUE
0132 408 :CLEAN STACK
0132 409 :RESTORE LOCAL SYMBOL TABLE LISTHEAD
0132 410 :REENABLE CTRL/Y'S
0132 411 :
0132 412 :GET DESCRIPTOR (R1/R2) OF SYMBOL VALUE.
0132 413 :R2=0 ==> VALUE IS AN INTEGER
0132 414 :
0132 415 GET_VALUE:
0132 416 :MOVZBL SYM_T_SYMBOL(R3),R2 :GET LENGTH OF SYMBOL
0132 417 :MOVAB SYM_T_SYMBOL+1(R3),R2 :GET ADDRESS OF VALUE LENGTH
0132 418 :CMPB SYM_B_TYPE(R3),#SYM_K_BINARY :NUMERIC BINARY VALUE?
0132 419 :BNEQ 10$ :BRANCH IF NOT
0132 420 :MOVL (R2),R1 :GET LONGWORD BINARY VALUE
0132 421 :CLRL R2 :MARK NOT A STRING
0132 422 :BRB 20$

7E 38 AB 7D 0132 393 MOVQ PRC_Q_LOCAL(R11),-(SP)
00A0 CB DD 0136 394 PUSHL PRC_L_STACKPT(R11)
50 38 AB 9E 013A 395 5$: MOVAB PRC_Q_LOCAL(R11),R0
49 10 013E 396 BSBB DCL$SEARCHT
15 50 E8 0140 397 BLBS R0,10$
50 8E D0 0143 398 MOVL (SP)+,R0
74 A0 9F 0146 399 PUSHAB IDF_K_LENGTH(R0)
38 AB 0084 C0 7D 0149 400 MOVQ IDF_Q_LOCAL+IDF_K_LENGTH(R0),PRC_Q_LOCAL(R11)
E6 5E A0 00 E0 014F 401 BBS #IDF_V_INPOP,IDF_W_FLAG(R0),5$
50 D4 0154 402 CLRL R0
02 11 0156 403 BRB 20$
09 10 0158 404 10$: BSBB GET VALUE
8E D5 015A 405 20$: TSTL (SP)+
38 AB 8E 7D 015C 406 MOVQ (SP)+,PRC_Q_LOCAL(R11)
0160 407 ENABLE
05 0162 408 RSB
0163 409
0163 410
0163 411
0163 412
0163 413
0163 414
0163 415
52 0C A3 9A 0163 416 MOVZBL SYM_T_SYMBOL(R3),R2
52 0D A342 9E 0167 417 MOVAB SYM_T_SYMBOL+1(R3),R2
02 0A A3 91 016C 418 CMPB SYM_B_TYPE(R3),#SYM_K_BINARY
07 12 0170 419 BNEQ 10$
51 62 D0 0172 420 MOVL (R2),R1
52 D4 0175 421 CLRL R2
03 11 0177 422 BRB 20$
```


SYMBOL
V04-000

M 5
- SYMBOL TABLE MANIPULATION ROUTINES
SEARCH FOR SYMBOL ENTRY IN A LOCAL SYMBO

16-SEP-1984 00:22:24 VAX/VMS Macro V04-00
4-SEP-1984 23:43:47 [DCL.SRC]SYMBOL.MAR;1

Page 12
(9)

51 82 3C 0179 423 10\$: MOVZWL (R2)+,R1 ;GET LENGTH OF VALUE
05 017C 424 20\$: RSB
017D 425

```
017D 427 .SBTTL SEARCH FOR SYMBOL ENTRY IN GLOBAL SYMBOL TABLE
017D 428 :+
017D 429 DCL$SEARCH_GLOBAL - SEARCH FOR SYMBOL ENTRY IN GLOBAL SYMBOL TABLE
017D 430 :
017D 431 THIS ROUTINE IS CALLED TO SEARCH THE GLOBAL SYMBOL TABLE FOR AN ENTRY.
017D 432 :
017D 433 INPUTS:
017D 434 :
017D 435 R11 = ADDRESS OF PROCESS WORK AREA
017D 436 :
017D 437 R1 = LENGTH OF SYMBOL.
017D 438 R2 = ADDRESS OF SYMBOL.
017D 439 :
017D 440 OUTPUTS:
017D 441 :
017D 442 R0 = STATUS
017D 443 R1/R2 = QUADWORD DESCRIBING SYMBOL VALUE:
017D 444 IF R2 NONZERO, QUADWORD IS A STRING DESCRIPTOR
017D 445 IF R2 ZERO, R1 IS A BINARY LONGWORD VALUE
017D 446 R3 = ADDRESS OF SYMBOL ENTRY.
017D 447 :
017D 448 :-
017D 449 :
017D 450 DCL$SEARCH_GLOBAL::
50 28 AB 9E 017D 451 MOVAB PRC_Q_GLOBAL(R11),R0 ;SEARCH FOR SYMBOL ENTRY IN GLOBAL TABLE
02 06 10 0181 452 BSBB DCL$SEARCHHT ;SET ADDRESS OF GLOBAL SYMBOL TABLE LISTHEAD
02 50 E9 0183 453 BLBC R0,10$ ;SEARCH GLOBAL SYMBOL TABLE FOR ENTRY
DB 10 0186 454 BSBB GET_VALUE ;IF LBC NO MATCH FOUND
05 0188 455 10$: RSB ;GET THE SYMBOL VALUE
0189 456 :
```



```
0189 458 .SBTTL SEARCH SPECIFIC SYMBOL TABLE FOR ENTRY
0189 459 :+
0189 460 : DCL$SEARCHT - SEARCH SPECIFIC SYMBOL TABLE FOR ENTRY
0189 461 :
0189 462 : THIS ROUTINE IS CALLED TO SEARCH A SPECIFIC SYMBOL TABLE FOR AN ENTRY.
0189 463 :
0189 464 : INPUTS:
0189 465 :
0189 466 : R0 = ADDRESS OF SYMBOL TABLE LISTHEAD.
0189 467 : R1 = LENGTH OF SYMBOL NAME.
0189 468 : R2 = ADDRESS OF SYMBOL NAME.
0189 469 :
0189 470 : OUTPUTS:
0189 471 :
0189 472 : R0 LOW BIT CLEAR INDICATES SEARCH FAILURE.
0189 473 :
0189 474 : R1 = LENGTH OF SYMBOL NAME.
0189 475 : R2 = ADDRESS OF SYMBOL NAME.
0189 476 : R3,R4 ARE DESTROYED.
0189 477 :
0189 478 : R0 LOW BIT SET INDICATES SYMBOL FOUND WITH:
0189 479 :
0189 480 : R1 = LENGTH OF SYMBOL NAME.
0189 481 : R2 = ADDRESS OF SYMBOL NAME.
0189 482 : R3 = ADDRESS OF SYMBOL ENTRY.
0189 483 : R4 = ADDRESS OF SYMBOL NAME STRING (JUST PAST THE COUNT).
0189 484 : -
0189 485 :
0189 486 DCL$SEARCHT::
0189 487 MOVAB PRC_Q_KEYPAD(R11),R3 :SEARCH SPECIFIC SYMBOL TABLE FOR ENTRY
0189 488 CMPL R0,R3 :GET ADDRESS OF KEYPAD TABLE
0189 489 BEQL 30$ :IS IT THE ONE WE WANT?
0189 490 MOVL R0,R3 :YES, THEN BRANCH
0189 491 10$: MOVL SYM_L_FL(R3),R3 :COPY ADDRESS OF SYMBOL TABLE LISTHEAD
0189 492 CMPL R0,R3 :GET ADDRESS OF NEXT ENTRY
0189 493 BEQL 20$ :END OF TABLE?
0189 494 MOVAB SYM_T_SYMBOL(R3),R4 :IF EQL YES
0189 495 CMPB R1,R4 :GET ADDRESS OF SYMBOL NAME
0189 496 BCTRU 10$ :SYMBOL LENGTH TOO LONG?
0189 497 SUBB3 SYM_B_NONUNIQUE(R3),(R4) :IF GTRU YES
0189 498 CMPB R1,R4 :SYMBOL LENGTH TOO SHORT?
0189 499 BLSSU 10$ :IF LSSU YES
0189 500 PUSHR #*M<R0,R1,R2,R3> :SAVE SEARCH PARAMETERS
0189 501 CMPC R1,(R2),(R4) :SYMBOLS MATCH?
0189 502 POPR #*M<R0,R1,R2,R3> :RESTORE SEARCH PARAMETERS
0189 503 BNEQ 10$ :IF NEQ NO
0189 504 INCL R0 :SET SUCCESS INDICATOR
0189 505 20$: RSB :
0189 506 :
0189 507 30$: BSBW DCL$FIND_KEYPAD :CALL KEYPAD SYMBOL SEARCH ROUTINE
0189 508 RSB :RETURN
```

53	40	AB	9E	0189	487	MOVAB	PRC_Q_KEYPAD(R11),R3	:SEARCH SPECIFIC SYMBOL TABLE FOR ENTRY
	53	50	D1	018D	488	CMPL	R0,R3	:GET ADDRESS OF KEYPAD TABLE
		2B	13	0190	489	BEQL	30\$:IS IT THE ONE WE WANT?
	53	50	D0	0192	490	MOVL	R0,R3	:YES, THEN BRANCH
	53	63	D0	0195	491	MOVL	SYM_L_FL(R3),R3	:COPY ADDRESS OF SYMBOL TABLE LISTHEAD
	53	50	D1	0198	492	CMPL	R0,R3	:GET ADDRESS OF NEXT ENTRY
		1F	13	019B	493	BEQL	20\$:END OF TABLE?
54	0C	A3	9E	019D	494	MOVAB	SYM_T_SYMBOL(R3),R4	:IF EQL YES
	64	51	91	01A1	495	CMPB	R1,R4	:GET ADDRESS OF SYMBOL NAME
		EF	1A	01A4	496	BCTRU	10\$:SYMBOL LENGTH TOO LONG?
7E	84	0B	83	01A6	497	SUBB3	SYM_B_NONUNIQUE(R3),(R4)	:IF GTRU YES
	8E	51	91	01AB	498	CMPB	R1,R4	:SYMBOL LENGTH TOO SHORT?
		E5	1F	01AE	499	BLSSU	10\$:IF LSSU YES
		0F	BB	01B0	500	PUSHR	#*M<R0,R1,R2,R3>	:SAVE SEARCH PARAMETERS
64	62	51	29	01B2	501	CMPC	R1,(R2),(R4)	:SYMBOLS MATCH?
		0F	BA	01B6	502	POPR	#*M<R0,R1,R2,R3>	:RESTORE SEARCH PARAMETERS
		DB	12	01B8	503	BNEQ	10\$:IF NEQ NO
		50	D6	01BA	504	INCL	R0	:SET SUCCESS INDICATOR
			05	01BC	505	RSB	:	:
				01BD	506	:	:	:
	FE40'	30	01BD	507	30\$: BSBW	DCL\$FIND_KEYPAD	:	:CALL KEYPAD SYMBOL SEARCH ROUTINE
		05	01C0	508	RSB	:	:	:RETURN


```
01C1 510 .SBTTL RESTORE SYMBOL DEFINITION AFTER A SPAWN
01C1 511 :+
01C1 512 DCL$RESTORE_SYM - RESTORE SYMBOL DEFINITION AFTER A SPAWN
01C1 513 :
01C1 514 THIS ROUTINE IS CALLED TO RESTORE A SYMBOL AFTER A SPAWN.
01C1 515 :
01C1 516 INPUTS:
01C1 517 :
01C1 518 R0 = SYMBOL TYPE
01C1 519 R1/R2 = DESCRIPTOR OF SYMBOL VALUE
01C1 520 R3/R4 = DESCRIPTOR OF SYMBOL NAME
01C1 521 R5 = ADDRESS OF SYMBOL TABLE LISTHEAD
01C1 522 R6 = ADDRESS OF CTX BLOCK
01C1 523 R11 = ADDRESS OF PROCESS WORK AREA
01C1 524 :
01C1 525 OUTPUTS:
01C1 526 :
01C1 527 THE SYMBOL IS ADDED TO THE TAIL OF THE SYMBOL TABLE.
01C1 528 :
01C1 529 R0 LOW BIT CLEAR INDICATES ALLOCATION FAILURE WITH CLIS_SYMOVF.
01C1 530 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
01C1 531 :-
01C1 532
01C1 533 DCL$RESTORE_SYM::
1F BB 01C1 534 PUSHRR #^M<R0,R1,R2,R3,R4> ;SAVE THE REGISTERS
01C3 535 :
01C3 536 :
01C3 537 : ALLOCATE THE SYMBOL.
01C3 538 :
51 0F C0 01C3 539 ADDL #SYM_T_SYMBOL+3,R1 ;GET SIZE OF SYMBOL NEEDED
50 02 91 01C6 540 CMPB #SYM_K_BINARY,R0 ;BINARY SYMBOL?
03 12 01C9 541 BNEQ 5$ ;NO, THEN SKIP
51 13 D0 01CB 542 MOVL #SYM_T_SYMBOL+3+4,R1 ;SET SIZE OF SYMBOL NEEDED
51 53 C0 01CE 543 5$: ADDL R3,RT ;
FF01 30 01D1 544 BSBW DCL$ALLDYNMEM ;ALLOCATE DYNAMIC MEMORY
43 50 E9 01D4 545 BLBC R0,90$ ;IF LBC ALLOCATION FAILURE
01D7 546 :
01D7 547 :
01D7 548 : INITIALIZE THE STATICALLY PLACED FIELDS AND INSERT IT IN THE LINKED LIST.
01D7 549 :
0B 08 A2 51 B0 01D7 550 MOVW R1,SYM_W_SIZE(R2) ;SET SIZE OF ALLOCATED BLOCK
A2 06 A6 90 01DB 551 MOVB CTX_B_NONUNIQUE(R6),SYM_B_FLAGS(R2) ;SET KEYPAD FLAGS
0A A2 6E 90 01E0 552 MOVB (SP),SYM_B_TYPE(R2) ;SET VALUE TYPE
6E 04 91 01E4 553 CMPB #SYM_K_KEYPAD,(SP) ;KEYPAD SYMBOL?
04 B5 62 0E 01E7 554 BNEQ 10$ ;NO, SKIP
65 62 0E 01E9 555 INSQUE (R2),04(R5) ;INSERT ENTRY AT TAIL OF TABLE
03 11 01ED 556 BRB 20$ ;SKIP
65 62 0E 01EF 557 10$: INSQUE (R2),(R5) ;INSERT ENTRY AT HEAD OF TABLE
01F2 558 :
01F2 559 :
01F2 560 : INITIALIZE THE DYNAMICALLY PLACED ASCII FIELDS.
01F2 561 :
53 0C AE 7D 01F2 562 20$: MOVQ 12(SP),R3 ;GET SYMBOL NAME
OC A2 53 90 01F6 563 MOVB R3,SYM_T_SYMBOL(R2) ;INSERT LENGTH OF SYMBOL
OD A2 64 53 28 01FA 564 MOVC R3,(R4),SYM_T_SYMBOL+1(R2) ;INSERT SYMBOL NAME
6E 02 91 01FF 565 CMPB #SYM_K_BINARY,(SP) ;BINARY VALUE
06 12 0202 566 BNEQ 30$ ;NO, THEN SKIP
```


SYMBOL
V04-000

D 6
- SYMBOL TABLE MANIPULATION ROUTINES
RESTORE SYMBOL DEFINITION AFTER A SPAWN

16-SEP-1984 00:22:24
4-SEP-1984 23:43:47

VAX/VMS Macro V04-00
[DCL.SRC]SYMBOL.MAR;1

Page 16
(12)

```

      83  04 AE  D0  0204  567      MOVL  4(SP),(R3)+      ;INSERT THE VALUE
      0A  11  0208  568      BRB      40$                ;SKIP
63    83  04 AE  B0  020A  569 30$:  MOVW  4(SP),(R3)+      ;INSERT LENGTH OF VALUE
08    BE  04 AE  28  020E  570      MOVC  4(SP),28(SP),(R3) ;INSERT SYMBOL VALUE
      1F  BA  0214  571 40$:  POPR   #^M<R0,R1,R2,R3,R4>    ;RESTORE THE REGISTERS
      50  01  D0  0216  572      MOVL  #1,R0              ;SET SUCCESS INDICATOR
      05  0219  573      RSB
      021A  574
      021A  575 ;
      021A  576 ; RETURN SYMBOL TABLE OVERFLOW STATUS.
      021A  577 ;
      1F  BA  021A  578 90$:  POPR   #^M<R0,R1,R2,R3,R4>    ;RESTORE THE REGISTERS
      021C  579      STATUS SYMOVF                        ;SET SYMBOL TABLE OVERFLOW STATUS
      05  0223  580      RSB
```

```
0224 582 .SBTTL DELETE SYMBOL FROM SYMBOL TABLE
0224 583 :+
0224 584 : DCL$DELSYM - DELETE SYMBOL FROM SYMBOL TABLE
0224 585 :
0224 586 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO DELETE EITHER A
0224 587 : SPECIFIC SYMBOL OR ALL SYMBOLS FROM A SPECIFIED SYMBOL TABLE.
0224 588 :
0224 589 : INPUTS:
0224 590 :
0224 591 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0224 592 : R9 = ADDRESS OF SCRATCH STACK.
0224 593 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0224 594 :
0224 595 : OUTPUTS:
0224 596 :
0224 597 : THE SPECIFIED SYMBOL, IF IT EXISTS, IS DELETED.
0224 598 : ELSE NO SUCH SYMBOL IS RETURNED;
0224 599 : OR THE ENTIRE CONTENTS OF THE SPECIFIED TABLE IS DELETED.
0224 600 : PERMANENT SYMBOLS ARE NEVER DELETED.
0224 601 :-
0224 602
0224 603 DCL$DELSYM::
0224 604     MOVAQ PRC_Q_LOCAL(R11),R7
0228 605     CLRL R8
022A 606     MOVAB LOCTAB,R9
022F 607 10$: BSBW DCL$GETDVAL
0232 608     CMPB #PTR_K_ENDLINE,R5
0235 609     BEQL 30$
0237 610     CMPB #PTR_K_PARAMETR,R5
023A 611     BEQL 30$
023C 612     CMPB #PTR_K_COMDQUAL,R5
023F 613     BNEQ 10$
0241 614     BSBW DCL$GETNVAL
0244 615     CMPB R1,#CLISK_DLSY_ALL
0248 616     BEQL 20$
024A 617     CMPB R1,#CLISK_DLSY_LOCA
024E 618     BEQL 15$
0250 619     CMPB R1,#CLISK_DLSY_GLOB
0254 620     BEQL 12$
0256 621     CMPB R1,#CLISK_DLSY_LOG
025A 622     BNEQ 10$
025C 623     BICB #1,R8
025F 624     BLBS R3,10$
0262 625     BISB #1,R8
0265 626     BRB 10$
0267 627 12$: MOVAQ PRC_Q_GLOBAL(R11),R7
026B 628     MOVAB GBLTAB,R9
0270 629     BRB 10$
0272 630 15$: MOVAQ PRC_Q_LOCAL(R11),R7
0276 631     MOVAB LOCTAB,R9
027B 632     BRB 10$
027D 633 20$: BISB #2,R8
0280 634     BRB 10$
0282 635 30$: BBS #1,R8,50$
0286 636     MOVL R7,R0
0289 637     BSBW DCL$SEARCHT
028C 638     BLBC R0,40$

57 38 AB 7E 0224 604 MOVAQ PRC_Q_LOCAL(R11),R7
58 D4 0228 605 CLRL R8
59 FDD2 CF 9E 022A 606 MOVAB LOCTAB,R9
FDCE' 30 022F 607 10$: BSBW DCL$GETDVAL
55 04 91 0232 608 CMPB #PTR_K_ENDLINE,R5
4B 13 0235 609 BEQL 30$
55 03 91 0237 610 CMPB #PTR_K_PARAMETR,R5
46 13 023A 611 BEQL 30$
55 00 91 023C 612 CMPB #PTR_K_COMDQUAL,R5
EE 12 023F 613 BNEQ 10$
FDBC' 30 0241 614 BSBW DCL$GETNVAL
00'8F 51 91 0244 615 CMPB R1,#CLISK_DLSY_ALL
33 13 0248 616 BEQL 20$
00'8F 51 91 024A 617 CMPB R1,#CLISK_DLSY_LOCA
22 13 024E 618 BEQL 15$
00'8F 51 91 0250 619 CMPB R1,#CLISK_DLSY_GLOB
11 13 0254 620 BEQL 12$
00'8F 51 91 0256 621 CMPB R1,#CLISK_DLSY_LOG
D3 12 025A 622 BNEQ 10$
58 01 8A 025C 623 BICB #1,R8
CD 53 E8 025F 624 BLBS R3,10$
58 01 88 0262 625 BISB #1,R8
C8 11 0265 626 BRB 10$
57 28 AB 7E 0267 627 12$: MOVAQ PRC_Q_GLOBAL(R11),R7
59 FD97 CF 9E 026B 628 MOVAB GBLTAB,R9
BD 11 0270 629 BRB 10$
57 38 AB 7E 0272 630 15$: MOVAQ PRC_Q_LOCAL(R11),R7
59 FD86 CF 9E 0276 631 MOVAB LOCTAB,R9
B2 11 027B 632 BRB 10$
58 02 88 027D 633 20$: BISB #2,R8
AD 11 0280 634 BRB 10$
15 58 01 E0 0282 635 30$: BBS #1,R8,50$
50 57 D0 0286 636 MOVL R7,R0
FEFD 30 0289 637 BSBW DCL$SEARCHT
04 50 E9 028C 638 BLBC R0,40$

: DELETE SYMBOL FROM TABLE
: ASSUME DELETING A SPECIFIC SYMBOL
: FROM THE LOC. TABLE WITH /NOLOG
:
: GET NEXT RESULT DESCRIPTOR
: IS THIS THE END OF THE LINE?
: BR IF YES
: IS THIS A PARAMETER?
: BR IF YES
: IS THIS A QUALIFIER?
: BR IF NO
: GET QUALIFIER NUMBER
: /ALL?
: BR IF ALL
: /LOCAL?
: BR IF LOCAL TABLE
: /GLOBAL?
: BR IF GLOBAL TABLE
: /LOG?
: IF NOT, IGNORE
: ASSUME /NOLOG
: IT IS /NOLOG. FLAG OK AS IS
: IT IS /LOG. SET FLAG ACCORDINGLY
: BR FOR MORE
: SET PROPER TABLE ADDRESS
: SET GLOBAL STRING ADDR. FOR /LOG MESS.
: BR FOR MORE
: MAKE SURE TABLE ADDRESS IS CORRECT
: SET LOCAL STRING ADDR. FOR /LOG MESS.
:
: SET DOING ALL FLAG
:
: BR IF DOING ALL
: SET TABLE ADDRESS
: SEARCH THE TABLE FOR THE SYMBOL
: DEALLOCATE
```



```

15 10 028F 639 BSBB 55$ : TEST AND DEALLOCATE
33 11 0291 640 BRB 70$ :
      05 0293 641 40$: STATUS UNDSYM : NO SUCH SYMBOL
53 67 D0 029A 642 MOVL (R7),R3 : EXIT WITH ERROR STATUS
53 57 D1 029E 643 50$: CMPL R7,R3 : GET ADDRESS OF NEXT SYMBOL
      23 13 02A1 645 BEQL 70$ : DONE?
      F5 AF 9F 02A3 646 PUSHAB 50$ : BR IF YES
01 0A A3 91 02A6 647 55$: CMPB SYM_B_TYPE(R3),#SYM_K_PERM : RETURN ADDRESS FOR DEALLOCATE
      18 13 02AA 648 BEQL 60$ : PERMANENT SYMBOL?
      12 58 E9 02AC 649 BLBC R8,59$ : BR IF YES - DON'T DEALLOCATE IT
      0C A3 9F 02AF 650 PUSHAB SYM_T_SYMBOL(R3) : SKIP IF /NOLOG
      59 DD 02B2 651 PUSHL R9 : GET SYMBOL NAME
50 51 02 02B4 652 MOVL #2,R1 : SET GLOBAL/LOCAL ASCII ADDR.
      0003DE23 8F D0 02B7 653 MOVL #CLIS_DELSYM,R0 : SET FAO COUNT TO 2
      FD3F' 30 02BE 654 BSBW DCL$FORMMSG : SET MESSAGE STATUS
      FE21 31 02C1 655 59$: BRW DCL$DEALLOCSYM : OUTPUT THE MESSAGE
      8E D5 02C4 656 60$: TSTL (SP)+ : DEALLOCATE IT
      05 02C6 657 70$: STATUS NORMAL : CLEAR INTERMEDIATE RETURN
      02CD 658 RSB :
      02CE 659 :
      02CE 660 .END
```

SYMBOL
Symbol table

- SYMBOL TABLE MANIPULATION ROUTINES

16-SEP-1984 00:22:24 VAX/VMS Macro V04-00
4-SEP-1984 23:43:47 [DCL.SRC]SYMBOL.MAR;1

Page 19
(13)

```

ABSYMD          000000BF R      02
ALLOCSYM        0000003B R      02
CLISK_DLSY_ALL  ***** X      02
CLISK_DLSY_GLOB ***** X      02
CLISK_DLSY_LOCA ***** X      02
CLISK_DLSY_LOG  ***** X      02
CLIS_ABSYMD     = 000381A0
CLIS_DELSYM     = 0003DE23
CLIS_NORMAL     = 00030001
CLIS_SYMABR     = 00038278
CLIS_SYMOVF     = 00038138
CLIS_UNDSYM     = 00038140
CTX_B_ACMODE    00000004
CTX_B_CONTINUE 00000012
CTX_B_FLAGS     0000000E
CTX_B_KEYLENGTH 00000002
CTX_B_NFLAGS    00000005
CTX_B_NONUNIQUE 00000006
CTX_B_PROMPTLEN 0000000F
CTX_B_SYMTAB     00000004
CTX_B_SYMTYPE    00000005
CTX_B_TFLAGS     00000005
CTX_B_TRANCNT    00000006
CTX_C_HDRLEN     00000033
CTX_G_PROMPT     00000013
CTX_K_HDRLEN     00000033
CTX_L_OUTOFBAND 0000000A
CTX_L_QUOTA      00000008
CTX_Q_PROCPRIV   00000002
CTX_T_CMDSTR     00000002
CTX_T_KEYSTATE   00000003
CTX_T_LNMNAME    00000007
CTX_T_LNMTABLE   0000000C
CTX_T_LOGNAM     00000005
CTX_T_SYMBOL     00000007
CTX_W_ENTSIZE    00000002
CTX_W_PMPTCTRL   00000010
CTX_W_PROT       00000006
CTX_W_TYPE       00000000
DCL$ALLDYNMEM    000000D5 RG     02
DCL$ALLOCSYM     00000025 RG     02
DCL$ALLOCSYMBR   0000000F RG     02
DCL$CBTA_DEC     ***** X      02
DCL$CVT_STRING   00000104 RG     02
DCL$DEADYNMEM    000000EC RG     02
DCL$DEALLOCSYM   000000E5 RG     02
DCL$DELSYM       00000224 RG     02
DCL$DISABLE      ***** X      02
DCL$FIND_KEYPAD  ***** X      02
DCL$FORMMSG      ***** X      02
DCL$GETDVAL      ***** X      02
DCL$GETNVAL      ***** X      02
DCL$GT_SYMBR     0000000D RG     02
DCL$RESTORE_SYM  000001C1 RG     02
DCL$SEARCH       00000117 RG     02
DCL$SEARCHT      00000189 RG     02
DCL$SEARCH_GLOBAL 0000017D RG     02

```

```

DCL$SEARCH_LOCAL 0000012C RG     02
DCL$SYM_STRING    000000FC RG     02
EXE$ALLOCATE      ***** X      02
EXE$DEALLOCATE    ***** X      02
GBLTAB            00000006 R      02
GET_VALUE         00000163 R      02
IDF_B_OUTFLAGS    00000038
IDF_C_LENGTH      00000074
IDF_K_LENGTH      00000074
IDF_L_FILENAME    00000068
IDF_L_INPRABCTX   0000000C
IDF_L_LNK         00000000
IDF_L_ONCTLY      00000060
IDF_L_ONERROR     00000008
IDF_L_OUTRABCTX   00000024
IDF_L_SEARCHCTX   00000064
IDF_Q_LABEL       00000018
IDF_Q_LOCAL       00000010
IDF_T_INPDVI      0000003C
IDF_T_OUTDVI      00000028
IDF_V_INPOP      = 00000000
IDF_W_FLAG        0000005E
IDF_W_INPDID      00000052
IDF_W_INPFID      0000004C
IDF_W_INPFI       00000004
IDF_W_INPRFA      00000058
IDF_W_ONLEVEL     00000006
IDF_W_OUTIFI      00000020
IDF_W_OUTISI      00000022
LOCTAB            00000000 R      02
PRC_B_CONTINUE    000000F3
PRC_B_DEFRADIX    000000AE
PRC_B_EXMDEPMOD   000000AD
PRC_B_EXMDEPWID   000000AC
PRC_B_EXONLYL     0000012D
PRC_B_FLAGS2      000000AF
PRC_B_IMGFLAG     00000078
PRC_B_OUTFLAGS    0000012C
PRC_B_PROMPTLEN   000000F0
PRC_C_LENGTH      00000534
PRC_G_COMMANDS    00000133
PRC_G_PROMPT      000000F4
PRC_K_LENGTH      00000534
PRC_L_CURRKEY     00000048
PRC_L_EXMDEPADR   000000A8
PRC_L_EXTARG      00000094
PRC_L_EXTBLK      0000008C
PRC_L_EXTCOD      0000009C
PRC_L_EXTHND      00000090
PRC_L_EXTPRM      00000098
PRC_L_IDFLNK      000000BC
PRC_L_IMGACTSTS   00000080
PRC_L_INDCLOCK    0000007C
PRC_L_INDEPTH     0000005C
PRC_L_INDFAB      0000001C
PRC_L_INDINPRAB   00000014
PRC_L_INDOUTRAB   00000018

```


SYMBOL
Symbol table

H 6
- SYMBOL TABLE MANIPULATION ROUTINES

16-SEP-1984 00:22:24 VAX/VMS Macro V04-00
4-SEP-1984 23:43:47 [DCL.SRC]SYMBOL.MAR;1

Page 20
(13)

```

PRC_L_INPRAB      00000008
PRC_L_LASTKEY     0000004C
PRC_L_LSTSTATUS   000000B0
PRC_L_ONCTLY      000000B8
PRC_L_ONERROR     0000006C
PRC_L_OUTOFBAND   000000B4
PRC_L_OUTRAB      0000000C
PRC_L_OUTRABCTX   00000118
PRC_L_PPFLIST     00000070
PRC_L_RECALLPTR   0000012F
PRC_L_RESTART     00000058
PRC_L_SAVAP       00000000
PRC_L_SAVFP       00000004
PRC_L_SEVERITY    00000050
PRC_L_SPWN        000000C0
PRC_L_STACKLM     000000A4
PRC_L_STACKPT     000000A0
PRC_L_STATUS      00000054
PRC_L_STS         00000084
PRC_L_STV         00000088
PRC_L_SYMBOL      00000060
PRC_L_TMBX        00000074
PRC_L_TRMLIST     00000010
PRC_Q_ALLOCREG    00000020
PRC_Q_COMMAND     000000E0
PRC_Q_FLUSHTIME   000000D0
PRC_Q_GLOBAL      00000028
PRC_Q_IMAGENAME   000000D8
PRC_Q_KEYPAD      00000040
PRC_Q_LABEL       00000030
PRC_Q_LOCAL       00000038
PRC_Q_SAVEPRIV    000000E8
PRC_T_OUTDVI      0000011C
PRC_W_ASTIOSB     000000C6
PRC_W_ASTRETN     000000C8
PRC_W_ASTSTATUS   000000C4
PRC_W_ATTMBX      0000007A
PRC_W_FLAGS       00000068
PRC_W_INPCHAN     00000064
PRC_W_ONLEVEL     0000006A
PRC_W_OUTIFI      00000114
PRC_W_OUTISI      00000116
PRC_W_OUTMBXCHN   000000CA
PRC_W_OUTMBXREF   000000CE
PRC_W_OUTMBXSIZ   000000CC
PRC_W_PMPTCTRL    000000F1
PRC_W_WAITIOSB    00000066
PTR_B_LEVEL       00000004
PTR_B_NUMBER      00000005
PTR_B_PARMCNT     00000006
PTR_B_VALUE       00000000
PTR_C_LENGTH      0000000C
PTR_K_CMDQUAL     = 00000000
PTR_K_ENDLINE     = 00000004
PTR_K_LENGTH      0000000C
PTR_K_PARAMETER   = 00000003
PTR_L_DESCR       00000000

```

```

PTR_L_ENTITY      00000008
SYMOVF            000000C8 R    02
SYM_B_FLAGS       0000000B
SYM_B_NONUNIQUE   0000000B
SYM_B_TYPE        0000000A
SYM_K_BINARY      = 00000002
SYM_K_KEYPAD      = 00000004
SYM_K_PERM        = 00000001
SYM_L_BL          00000004
SYM_L_FL          00000000
SYM_T_SYMBOL      0000000C
SYM_W_SIZE        00000008
WRK_B_CMDOPT      FFFFFFFC3
WRK_B_MAXPARM     FFFFFFFD0
WRK_B_MINPARM     FFFFFFFD1
WRK_B_PARMCNT     FFFFFFFCE
WRK_B_PARMSUM     FFFFFFFCF
WRK_B_RECALLCNT   FFFFFFFC5
WRK_B_VALLEV      FFFFFFFC4
WRK_B_VERBTYP     FFFFFFFC2
WRK_C_LENGTH      FFFFF486
WRK_G_BUFFER      FFFFF492
WRK_G_INPBUF      FFFFF896
WRK_G_RESULT      FFFFF9B6
WRK_K_LENGTH      FFFFF486
WRK_L_CHARPTR     FFFFF48E
WRK_L_DISALLOW    FFFFFFFE6
WRK_L_ERRORRTN    FFFFF9AE
WRK_L_EXPANDPTR   FFFFF486
WRK_L_IMAGE       FFFFFFFE2
WRK_L_MARKPTR     FFFFF48A
WRK_L_PAROUT      FFFFFFFD2
WRK_L_PMPTADDR    FFFFF9A2
WRK_L_PROMPTRTN   FFFFF9A6
WRK_L_PROPTR      FFFFFFFC6
WRK_L_QUABLK      FFFFFFFCA
WRK_L_READRTN     FFFFF9AA
WRK_L_RECALLPTR   FFFFFFFEA
WRK_L_RSLEND      FFFFFFFB6
WRK_L_RSLNXT      FFFFFFFBA
WRK_L_SAVAP       FFFFFFFF8
WRK_L_SAVFP       FFFFFFFFC
WRK_L_SAVSP       FFFFFFFF4
WRK_L_SIGNALRTN   FFFFFFFD6
WRK_L_SPECRTN     FFFFF9B2
WRK_L_TAB_VEC     FFFFFFFDE
WRK_L_VERB        FFFFFFFBE
WRK_W_FLAGS       FFFFFFFF0
WRK_W_FLAGS2      FFFFFFFF2
WRK_W_IMGCHAN     FFFFFFFEE
WRK_W_PMPTLEN     FFFFF99E

```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes																
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE						
\$AB\$\$	FFFFFFFFC (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE						
DCL\$ZCODE	000002CE (718.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE						

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.05	00:00:01.48
Command processing	80	00:00:00.70	00:00:05.37
Pass 1	239	00:00:08.11	00:00:26.87
Symbol table sort	0	00:00:00.84	00:00:03.02
Pass 2	114	00:00:01.88	00:00:07.74
Symbol table output	25	00:00:00.17	00:00:01.17
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	469	00:00:11.80	00:00:45.68

The working set limit was 1350 pages.

39437 bytes (78 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 574 non-local and 34 local symbols.

660 source lines were read in Pass 1, producing 16 object records in Pass 2.

44 pages of virtual memory were used to define 24 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	11
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	15

737 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:SYMBOL/OBJ=OBJ\$:SYMBOL MSRC\$:SYMBOL/UPDATE=(ENH\$:SYMBOL)+EXECML\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0074 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

